

CTA Standard

OpenEPG™ – A Specification for Electronic Program Guide Data Interchange

CTA-2033 R2016

(Formerly ANSI/CEA-2033)

March 2008

NOTICE

Consumer Technology Association (CTA)™ Standards, Bulletins and other technical publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for his particular need. Existence of such Standards, Bulletins and other technical publications shall not in any respect preclude any member or nonmember of the Consumer Technology Association from manufacturing or selling products not conforming to such Standards, Bulletins or other technical publications, nor shall the existence of such Standards, Bulletins and other technical publications preclude their voluntary use by those other than Consumer Technology Association members, whether the standard is to be used either domestically or internationally.

Standards, Bulletins and other technical publications are adopted by the Consumer Technology Association in accordance with the American National Standards Institute (ANSI) patent policy. By such action, the Consumer Technology Association does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the Standard, Bulletin or other technical publication.

This document does not purport to address all safety problems associated with its use or all applicable regulatory requirements. It is the responsibility of the user of this document to establish appropriate safety and health practices and to determine the applicability of regulatory limitations before its use.

This document is copyrighted by the Consumer Technology Association (CTA)™ and may not be reproduced, in whole or part, without written permission. Federal copyright law prohibits unauthorized reproduction of this document by any means. Organizations may obtain permission to reproduce a limited number of copies by entering into a license agreement. Requests to reproduce text, data, charts, figures or other material should be made to the Consumer Technology Association (CTA)™.

(Formulated under the cognizance of the CTA **R7 Consumer Electronics Networking Committee.**)

Published by
©CONSUMER TECHNOLOGY ASSOCIATION 2016
Technology & Standards Department
www.cta.tech

All rights reserved

FOREWORD

This standard was developed under the auspices of the Consumer Technology Association (CTA) R7 Consumer Electronics Networking Committee.

Table of Contents

1 Introduction and Scope.....	1
1.1 Scope and Purpose	1
1.2 General Capabilities	1
1.3 Background and Usage Models	2
1.4 Principles of Operation	4
2 References	6
2.1 Normative References	6
2.2 Informative References	7
2.3 Reference Acquisition	7
3 Conventions and Definitions	8
3.1 Conventions.....	8
3.2 Definitions.....	8
3.2.1 Abbreviations/Acronyms	9
4 Requirements (Normative)	11
4.1 Data Elements	11
4.1.1 Codes for Indicating the Language of Content Streams	11
4.1.2 Basic Data Hierarchy	11
4.1.3 Structure of Elements.....	12
4.1.4 List of Data Elements.....	18
4.1.5 Externally Defined Data Types.....	18
4.1.6 Expansion of Element Types	19
4.1.7 OpenEPG Namespace.....	19
4.1.8 OpenEPG Transport Namespace	19
4.2 Communications	20
4.2.1 Discovery Functions	20
4.2.1.1 WAN Discovery	20
4.2.1.2 Referencing a UDDI Directory	21
4.2.1.3 Referencing a WSDL Implementation	21
4.2.1.4 LAN Discovery.....	23
4.2.2 Security Validity and Authentication Functions.....	23
4.2.3 MetaData Access Functions.....	23
4.2.3.1 Overview	23
4.2.3.2 Operation of a Data Request.....	24
4.2.3.3 Queries.....	24
4.2.3.3.1 get_Capabilities: Establishing the queries that an MSP supports	25
4.2.3.3.2 get_Data: Metadata retrieval	26
4.2.3.3.3 Responses to Queries	29
4.2.3.3.4 Sort Order of Elements.....	30
4.2.3.3.5 Trimming of the Response Document	31
4.2.4 Request Templates	31
4.2.4.1 DistributionNetworkId	32
4.2.4.2 GeneralSchedule.....	32
4.2.4.3 ScheduleUpdates	33
4.2.4.4 TitleSearch.....	34
4.2.4.5 DescriptionSearch.....	34

4.2.4.6 ContentID	34
4.2.4.7 ServiceID	35
4.2.5 Response Templates	35
4.2.5.1 LastDistributionNetworkUpdate	35
4.2.5.2 ServiceList	35
4.2.5.3 ServiceInformation	37
4.2.5.4 GeneralScheduleInformation	37
4.2.5.5 FullScheduleInformation	39
4.2.5.6 ContentDetail	42
4.2.6 Notification Functions	44
4.2.6.1 Notification Methods (Informative)	44
4.2.6.2 Update Tracking	45
4.2.6.2.1 LastUpdateTime for Distribution Network	45
4.2.6.2.2 ServiceLastUpdateTime for the ContentServiceSource	45
4.2.6.2.3 LastScheduleUpdate for the Event List	45
4.2.7 Application notes for use of Request and Response Templates (Informative)	46
4.2.8 General notes: Data Compression	48
5 Annexes	49
5.1 Annex A: Detailed Element Definitions (Normative)	50
5.1.1 AudioAttributesType	51
5.1.2 AVAttributesType	52
5.1.3 CallSignType	52
5.1.4 ContentType	53
5.1.5 ContentAdvisoryType	65
5.1.6 ContentServiceDistributionMethodType	65
5.1.7 ContentServiceMappingType	66
5.1.8 ContentServiceSourceType	68
5.1.9 ContentServiceType	69
5.1.10 ContentTitleType	69
5.1.11 DeviceType	70
5.1.12 DistributionNetworkType	70
5.1.13 EnhancementType	71
5.1.14 EpisodeNumberType	72
5.1.15 EpisodeTitleType	72
5.1.16 EventType	73
5.1.17 ExtendedLanguageType	75
5.1.18 ExtendedSubtitleCaptionType	76
5.1.19 GranularityType	76
5.1.20 GraphicFormatType	76
5.1.21 GraphicType	76
5.1.22 Language639-2Type	77
5.1.23 LocationCodeType	77
5.1.24 MetadataOriginCodeTableType	77
5.1.25 MetadataOriginItemType	78
5.1.26 MetadataOriginCodeItemBaseType	78
5.1.27 NetworkTypeType	78

5.1.28 NumberOfItemsType	78
5.1.29 OpenEPGType	79
5.1.30 OpenEPGBaseType	79
5.1.31 PartNumberType	80
5.1.32 PPVType	80
5.1.33 ProgramFormatType	80
5.1.34 ProgramReviewType	81
5.1.35 RecordRightsType	81
5.1.36 RightsType	81
5.1.37 RightsXMLType	82
5.1.38 ScheduleUpdateType	83
5.1.39 SegmentListType	83
5.1.40 ShortDescriptionType	84
5.1.41 ShortTitleType	84
5.1.42 SubscriptionBundleType	84
5.1.43 SubtitleCaptionTechniqueType	85
5.1.44 SubtitleCaptionType	85
5.1.45 TextualOriginType	85
5.1.46 TextualType	85
5.1.47 UDefIDType	86
5.2 Annex B: Detailed Transport Schema Element Definitions (Normative)	87
5.2.1 AvailableRequestTemplatesListType	87
5.2.2 AvailableResponseTemplatesListType	87
5.2.3 AvailableRequestTemplatesType	87
5.2.4 AvailableResponseTemplatesType	87
5.2.5 ExtendedFieldType	88
5.2.6 get_Capabilities_ResultType	88
5.2.7 get_CapabilitiesType	88
5.2.8 get_Data_ResultType	89
5.2.9 get_DataType	89
5.3 Annex C: Implementation Guidelines (Informative)	91
5.4 Annex D: OpenEPG XML Schemas (Informative)	94

Figures

Figure 1 - Typical Direct Access Scenario	2
Figure 2 – Typical Indirect Access Scenario	3
Figure 3 – Flow of Direct and Indirect Access Scenarios	5

OpenEPG™¹ - A specification for Electronic Program Guide Data Interchange

1 Introduction and Scope

1.1 Scope and Purpose

The OpenEPG™ standard defines a field structure and access method for obtaining electronic program guide (EPG) data, also known as metadata, for describing audio-video content and its availability using IP-related protocols. The OpenEPG standard facilitates access by home entertainment devices to scheduled event data for terrestrial, cable and satellite programming; to video on demand (VOD) services; and to content stored locally on a home networked device. OpenEPG allows for aggregation of metadata provided by multiple sources such as various metadata service providers (MSPs,) including small and local MSPs (such as churches, schools, etc.), and personal metadata for content provided by a user.

This standard defines the format and structure of OpenEPG data fields, and it defines methods of querying OpenEPG metadata services to request subsets of the available data. Electronic versions of the complete OpenEPG XML schemas and example files can be obtained from CEA [24].

This standard leverages the field structures used in the TVAnytime field set, but extends their use for the North American broadcast, cable and satellite infrastructure and conventions. This standard uses other common field structures, such MPEG-7, to the fullest extent possible.

1.2 General Capabilities

This standard enables the following capabilities:

- Enables devices to communicate using messages based upon IP protocols.
- Provides mechanisms that enable applications to tune to content.
- Provides mechanisms that enable applications to record content.
- Limits complexity to allow small providers to create and define program metadata elements.
- Provides alignment with existing DLNA architecture for in-home network device interaction.
- Provides methods of filtering EPG data for a particular geographical area (“localization”)
- Provides optional methods for user and/or device authentication and secure data communications with the EPG MSP.

The following capabilities have been identified for future consideration.

OpenEPG™ is a trade mark owned by the Consumer Electronics Association (CEA). Use of this term in descriptions of products or services conforming to this standard is permissible as long as it is accompanied by attribution.

- Methods that facilitate alternate business models for EPG MSPs, e.g. a method of preserving advertising content.

1.3 Background and Usage Models

The OpenEPG specification enables the exchange of media-content metadata between MSPs and CE equipment, and between devices on a home network through standardized formats and protocols. The goal of the OpenEPG initiative is to stimulate innovation in the applications and user experiences that utilize this metadata and to encourage the development of a robust set of products and services that can help users navigate and enjoy the increasingly diverse media content that is now becoming available to them.

In formulating the requirements for this specification, two classes of OpenEPG use-case were considered – Direct Access and Indirect Access. Figure 1 illustrates a typical Direct Access scenario:

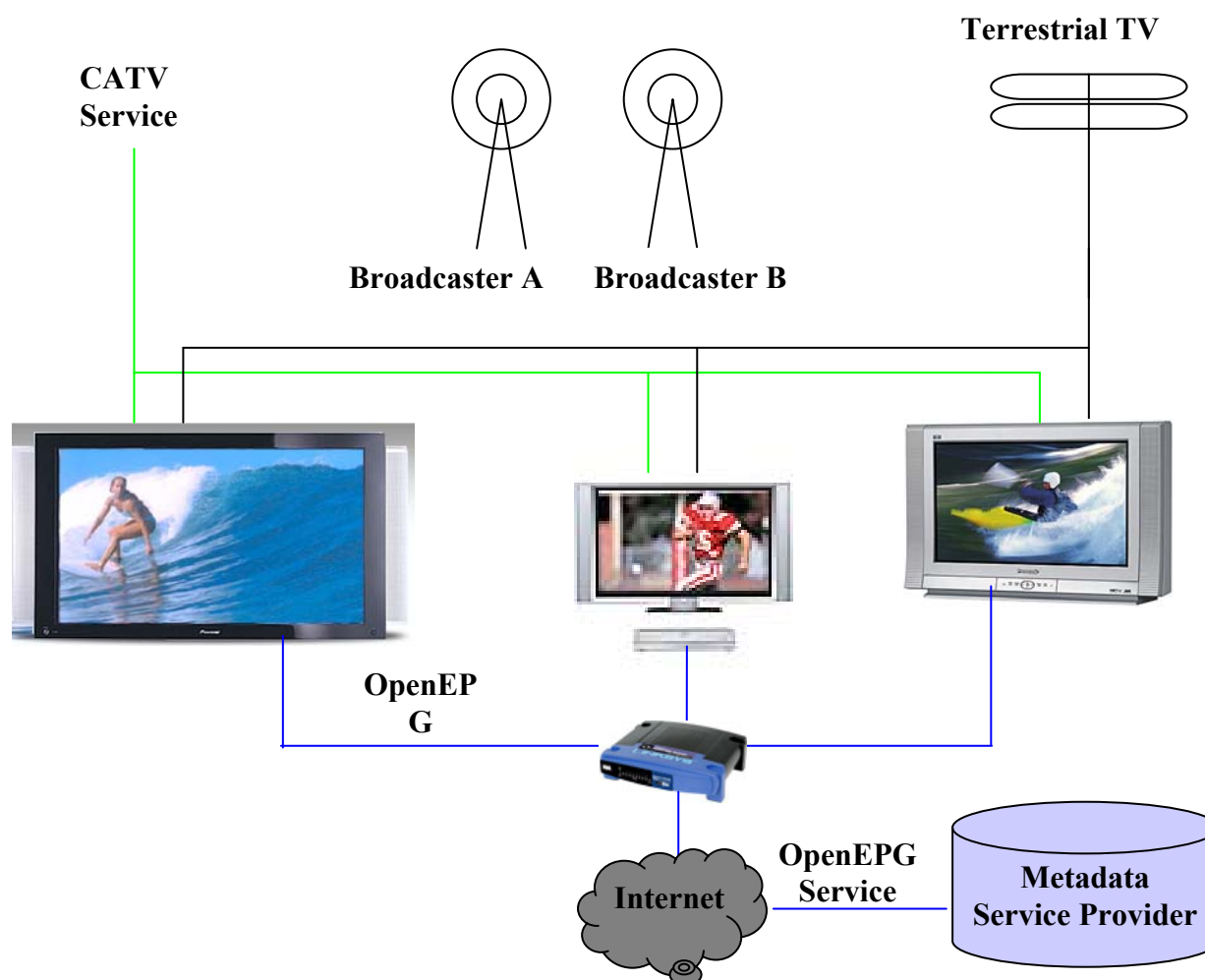


Figure 1 - Typical Direct Access Scenario

The user owns multiple digital televisions (DTVs), subscribes to a Cable TV service, and receives over-the-air television programming from multiple broadcasters. The DTVs are IP-enabled and are connected to a broadband Internet service via a home network. The user has also subscribed to an OpenEPG metadata service provided by a Metadata Service Provider (MSP).

The user can access the MSP service directly from each DTV in the home. (Subscription mechanisms are outside the scope of this standard.)

The MSP's OpenEPG service delivers detailed EPG metadata for content available from the Cable TV service or the OTA broadcasts. Additional content sources, delivery technologies, and types of CE device can be accommodated, and the user may subscribe to more than one OpenEPG metadata service. (In this case, another application would aggregate the metadata for display to the user).

The user browses the received EPG metadata to make current and future viewing decisions, and may use it to facilitate scheduled recording of selected programs.

Figure 2 shows a typical Indirect Access scenario:

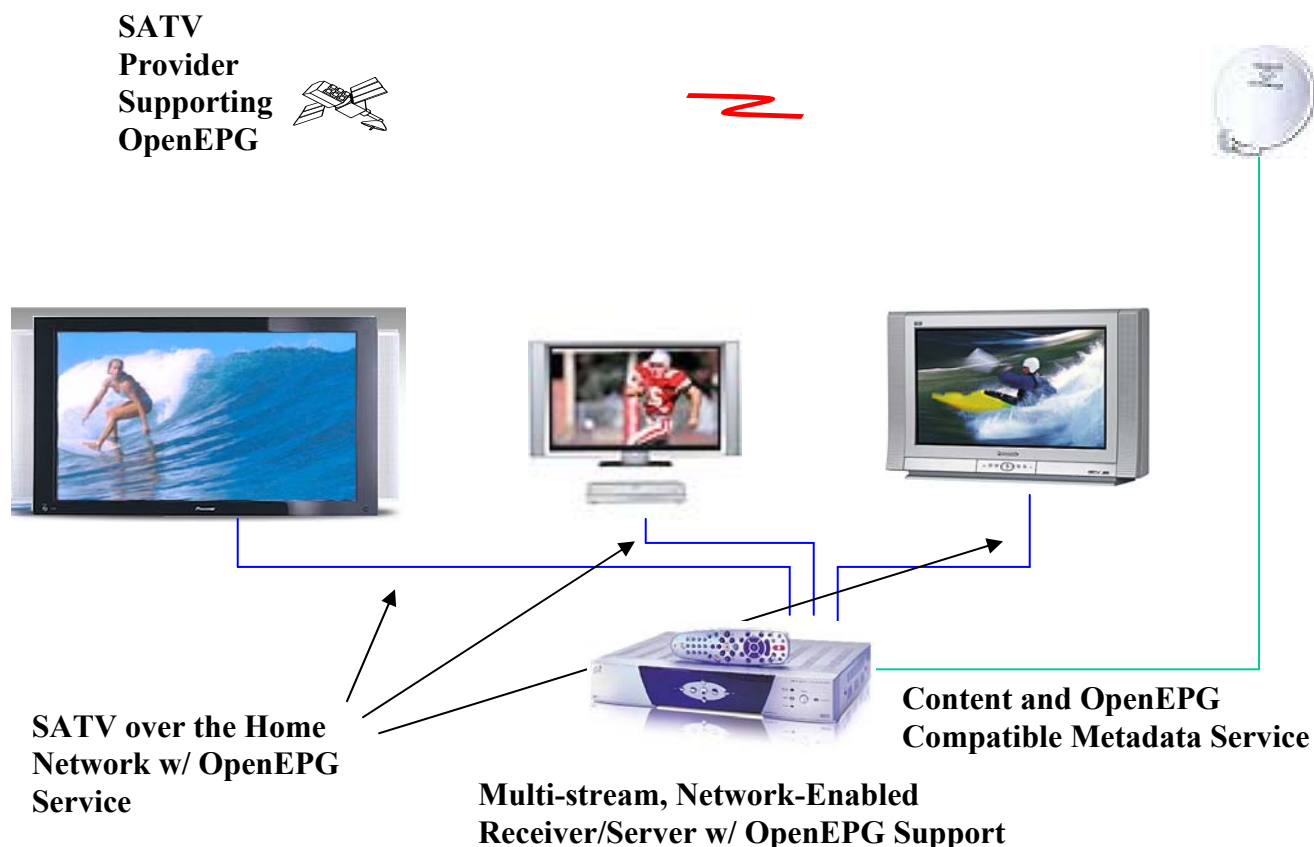


Figure 2 – Typical Indirect Access Scenario

In this particular scenario, the user owns multiple digital televisions (DTVs), and subscribes to a digital satellite service (DSS) that supports home networking features and OpenEPG. The DTVs are IP-enabled and are connected to the DSS receiver via the home network.

Both media content and metadata flows from the service provider to the DSS receiver and is delivered to each of the DTVs via IP transport over the home network. The user browses the metadata to make current and future viewing decisions and optionally uses the metadata to facilitate scheduled recording of selected programs. The OpenEPG metadata service provides event information as well as detailed metadata. Note that additional content receivers (renderers and/or storage devices), delivery technologies (additional satellite, OTA, or cable), and other types of CE devices could also be accommodated, and the user could receive metadata from more than one MSP. In the latter case, an application would aggregate the metadata for display to the user.

Scenarios that combine direct and indirect access methodologies can also be envisioned where applications on the CE device aggregate the metadata to provide a rich user experience.

1.4 Principles of Operation

See Figure 3 for an illustration of how OpenEPG is used in both direct and indirect access scenarios.

Copyright Consumer Technology Association

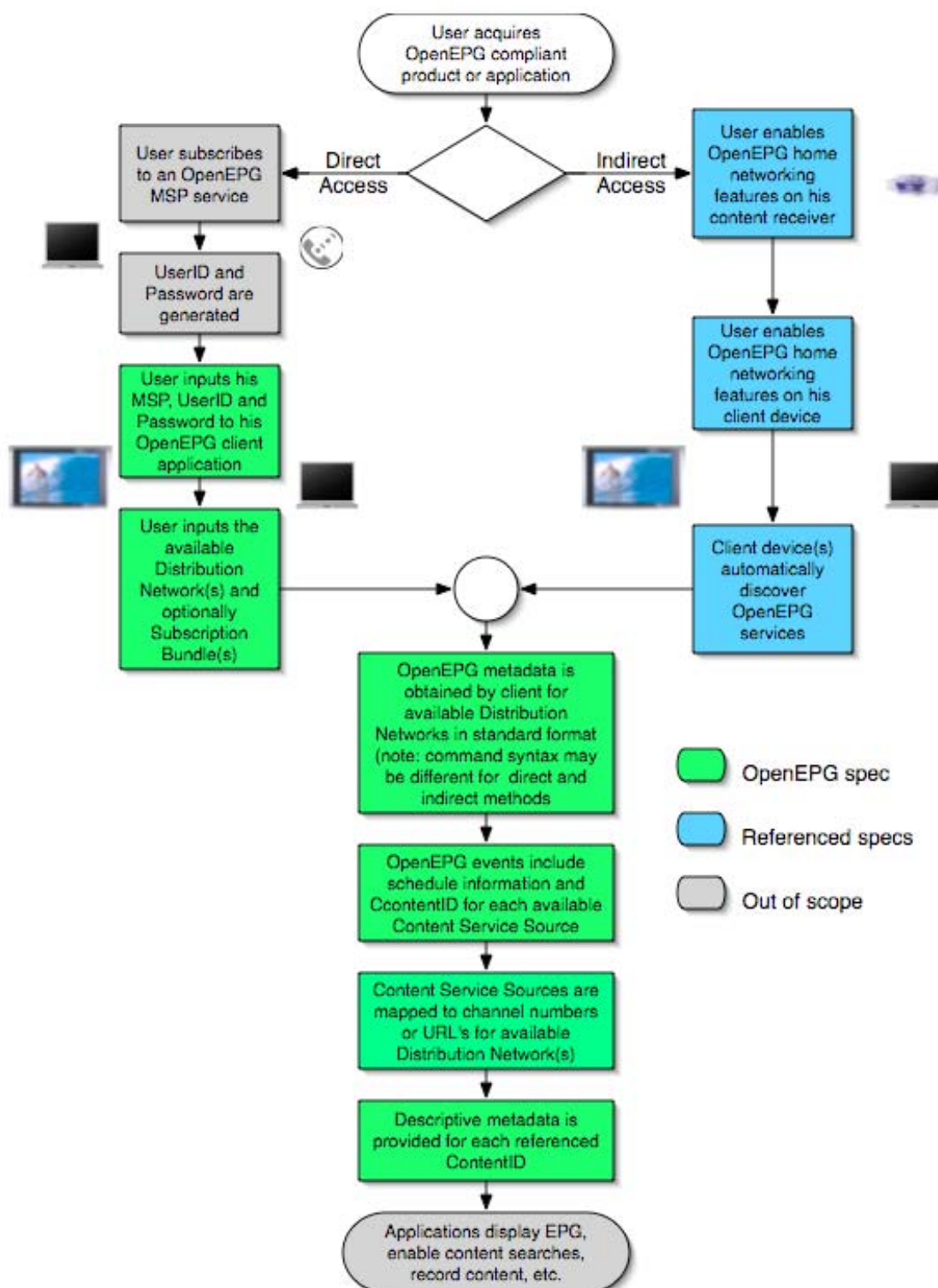


Figure 3 – Flow of Direct and Indirect Access Scenarios

2 References

2.1 Normative References

The following standards contain provisions that, through reference in this text, constitute normative provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed.

- [1] ATSC Standard A/65C Program and System Information Protocol for Terrestrial Broadcast and Cable, Rev. C, with Amendment No. 1, May 9, 2006
(http://www.atsc.org/standards/a_65cr1_with_amend_1.pdf)
- [2] CEA-CEB16 Active Format Description (AFD) & Bar Data Recommended Practice, July 2006
- [3] CEA-608-D Line 21 Data Services, May 2007
- [4] DLNA Networked Device Interoperability Guidelines, expanded: March 2006, Volume 1: Architectures and Protocols (DLNA Expanded Guidelines, Architectures and Protocols) and Volume 2: Media Format Profiles (DLNA Expanded Guidelines, MF)DLNA Home Networked Device Interoperability Guidelines v1.0
- [5] ETSI TS 102 822 Broadcast and Online Services: Search, Select and Rightful Use of Content on Personal Storage Systems ("TV-Anytime"), January 2006
- [6] ETSI TS 102 822-2 Title: Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 2: System Description, January 2006
- [7] ETSI TS 102 822-3-1 Title: Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 3: Metadata; Sub-part 1: Phase 1 - Metadata schemas, January 2006
- [8] ETSI TS 102 822-6-1 Title: Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime Phase 1"); Part 6: Delivery of metadata over a bi-directional network; Sub-part 1: Service and transport, January 2006
- [9] ETSI TS 102 822-6-2 (2006-01) Title: Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime Phase 1"); Part 6: Delivery of metadata over a bi-directional network; Sub-part 2: Service discovery, January 2006
- [10] IETF RFC 2616, HyperText Transfer Protocol – HTTP 1.1, June 1999
- [11] IETF RFC 2617 HTTP Authentication: Basic Digest Access Authentication, June 1999
- [12] IETF RFC 4646 Tags for Identifying Languages, February 2006
- [13] IETF RFC 4078 The TV-Anytime Content Reference Identifier (CRID), May 2005
- [14] IETF RFC 4246 International Standard Audio Visual Number (ISAN) URN Definition, February 2006
- [15] ISO 15706:2002 Information and Documentation - International Standard Audiovisual Number (ISAN)
- [16] ISO 639-1:2002 Codes for the representation of names of languages - Part 1: Alpha-2 code,

- [17] ISO 639-2:2007 Codes for the representation of names of languages - Part 2: Alpha-3 code for comprehensive coverage of languages
- [18] ISO/IEC 15938:2005 Information technology - Multimedia Content Description Interface, Parts 1-10 (MPEG-7)
- [19] ISO/IEC 15938-5:2003, Information Technology - Multimedia Content Description Interface - Part 5: Multimedia Description Schemes.
- [20] W3C Recommendation 24 December 1999, HTML 4.01 specification,
<http://www.w3c.org/TR/1999/REC-html401-19991224>
- [21] W3C Recommendation 24 June 2003, SOAP Version 1.2 Part 1: Messaging Framework
- [22] “Extensible Markup Language (XML) 1.0 (Fourth Edition) –
W3C Recommendation 16 August 2006, edited in place 29 September 2006”
<http://www.w3.org/TR/2006/REC-xml-20060816/>

2.2 Informative References

- [23] ETSI EN 300 468, Edition: 1.7.1(2006-05) Specification for Service Information (SI) in DVB systems (DVB SI)
(http://webapp.etsi.org/action/PU/20060523/en_300468v010701p.pdf)
- [24] Electronic versions of the complete OpenEPG XML Schemas and Example Files

2.3 Reference Acquisition

ATSC Standards:

- 1750 K Street, NW, Suite 1200, Washington DC, 20006 USA, Phone: +1-202-872-9160, Fax: +1-202-872-9161, Internet: Email: <http://www.atsc.org/>, atsc@atsc.org

CEA Standards:

- Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet <http://global.ihs.com> ; E-mail global@ihs.com

CEA-2033 OpenEPG XML Schemas and Example files:

- Consumer Electronics Association, 1919 South Eads Street, Arlington, VA USA 22202; Phone 703-907-7625; Fax 703-907-7693; Internet www.CE.org ; E-mail standards@CE.org

DLNA Documents:

- Digital Living Network Alliance (DLNA) Administration, C/O VTM, Attn: Membership Services. 5440 SW Westgate Drive, Suite217, Portland, OR 97221 USA Phone: 503.297.0426, Fax: 503.297.1090, Internet: <http://www.dlna.org/> , Email: admin@dlna.org

ETSI Standards:

- ETSI Secrétariat, 650, route des Lucioles, 06921 Sophia-Antipolis Cedex, FRANCE, Tel.: +33 (0)4 92 94 42 00, Fax: +33 (0)4 93 65 47 16, Internet: <http://www.etsi.org/#>, Email: helpdesk@etsi.org

IETF Documents:

- Internet Engineering Task Force (IETF) Secretariat, c/o Corporation for National Research Initiatives, 1895 Preston White Drive, Suite 100, Reston, VA 20191-5434, USA; Phone 703- 620-8990; Fax 703-620-9071; Internet www.ietf.org, IETF RFCs may be downloaded from www.ietf.org/rfc.html, IETF Internet drafts may be downloaded from www.ietf.org/ID.html

ISO Standards

- ISO Central Secretariat: International Organization for Standardization (ISO); 1, rue de Varembe, Case postale 56; CH-1211 Geneva 20, Switzerland; Telephone +41 22 749 01 11; Fax +41 22 733 34 30; <http://www.iso.org>

W3C Documents:

- World Wide Web Consortium (W3C); Internet: <http://www.w3.org/>

3 Conventions and Definitions

3.1 Conventions

Several keywords are used to differentiate levels of requirements and optionality, as follows:

MAY: A keyword that indicates flexibility of choice with no implied preference.

SHALL: A keyword that indicates a mandatory requirement. Designers are required to implement all such mandatory requirements to assure interoperability with other products conforming to this standard.

SHOULD: A keyword that denotes flexibility of choice with a strongly preferred alternative. Equivalent to the phrase “is recommended.”

Occurs: Mandatory elements defined in types, that is Occurs equals 1 or more, are dependent upon the use of the type in another element. A child element is only mandatory if its type is instantiated in another element where actual data exists.

In addition XML [22] establishes mandatory requirements that are equivalent to SHALL.

3.2 Definitions

For the purposes of this standard, the following definitions apply:

Content Service Provider (CSP): A service that provides actual Audio/Video content. The service MAY also provide MetaData related to the content they provide.

General Queries: Any OpenEPG data query. Particularly that does not conform to one of the predefined Template Queries. A Metadata Service Provider that indicates that it will accept General Queries SHALL accept all legal query expressions.

Metadata Service Provider (MSP): A service that provides MetaData for Audio/Video content provided by other services.

OpenEPG – A standard defining a language and access protocol for expressing Electronic Program Guide information.

OpenEPG Web Services – OpenEPG access protocols over the WAN (?)

Request Template – a predefined query format for data requests.

Response Template – a predefined list of fields that MAY be returned by a data request.

Service – can come from MSP or CDS

3.2.1 Abbreviations/Acronyms

For the purposes of this standard, the following abbreviations/acronyms apply:

ANSI - American National Standards Institute
ATSC - Advanced Television Systems Committee
CATV - Cable TeleVision
CDS - Content Delivery Service
CE - Consumer Electronics
CRID - Content Reference ID
CSP - Content Service Provider
DDD - Device Description Documents
DLNA - Digital Living Network Alliance
DSMCC - Digital Storage Media Control and Command
DSS - Digital Satellite Service
DTV - Digital TeleVision
DVB - Digital Video Broadcast
DVB SI - Digital Video Broadcast System Information
EPG - Electronic Program Guide
ETSI - European Telecommunications Standards Institute
HTML - HyperText Markup Language [20]
HTTP - HyperText Transfer Protocol [10]

IEC - International Electrotechnical Commission
IETF - Internet Engineering Task Force
IP - Internet Protocol
ISAN - International Standard Audio/Video Number
ISO - International Standards Organization
LAN - Local-Area Network
MD5 - Message-Digest Algorithm 5
MPEG - Moving Picture Experts Group
MSP - Metadata Service Provider
OTA - Over-The-Air
PSIP - Program And System Information
RFC - Request For Comment
SOAP - Simple Object Access Protocol Version 1.1
TCP/IP - Transport Control Protocol / Internet Protocol
TLS - Transport Layer Security
TVA - "TV-Anytime"
UDDI - Universal Description, Discovery, and Integration
URL - Universal Resource Locator
URN - Universal Resource Name
UTC - Coordinated Universal Time
VOD - Video On Demand
W3C - World Wide Web Consortium
WAN - Wide Area Network
WSDL - Web Services Description Language
WSIL - Web Services Inspection Language
XHTML - Extensible HyperText Markup language
XML - Extensible Markup Language

4 Requirements (Normative)

The requirements section is divided into two parts.

Section 4.1 - “Data Elements” describes the structure and content of each Data Element defined by the standard.

Section 4.2 - “Communications” describes the Query mechanism for requesting these Data Elements, the transport protocols for transferring the Queries and Responses, and a set of Templates that describe Standard Queries and their returned data sets.

4.1 Data Elements

This section describes the full set of EPG data elements. An OpenEPG MSP MAY support a subset of these elements but clients SHALL be capable of recognizing every element, even if the element is subsequently ignored.

All data elements are defined as XML Types so that standard XML extension mechanisms MAY be used.

4.1.1 Codes for Indicating the Language of Content Streams

In some regional standards such as the ATSC standard for terrestrial broadcast in the US (and by inheritance, US cable), languages for audio streams and closed captioning are labeled using ISO 639-2/B three-character codes. While it would be possible for a receiver to include mapping tables to derive the three-character equivalent code if only the two-character ISO-639-1 code is available, to avoid this complexity CEA-2033 requires both forms for these regions. The requirement for definition of audio and subtitle/captioning streams in Oepg:ContentType uses the rules in RFC-4646 (use the shorter form if available) via the mpeg7:ExtendedLanguageType. When a given piece of content appears in the broadcast stream, descriptive metadata is given in Oepg:EventType and the languages of audio and subtitle/captions streams are given in Oepg:ExtendedLanguageType. This latter structure contains a pair of elements for fields that specify the language of streams. One of these element pairs SHALL specify the language using the ISO-639-2/B three character code lists. The other element in the element pair SHALL specify the language according to the rules in RFC-4646[12]

4.1.2 Basic Data Hierarchy

The basic high-level hierarchy for all OpenEPG data is shown in Section 4.1.2. In general all data has “OpenEPG” as its root element. Three major elements branch from this root element, which describe the Distribution Network, the Content Service Source, and the Content.

The Distribution Network Element provides information specific to a particular distribution method such as a Satellite network, Cable network or Internet Service or specific file server that distributes content. For OTA or broadcast-type networks it describes the mapping of specific broadcast services to the channel numbers for that particular distribution network.

The Content Service Source Element provides information about the originating broadcaster or service and the event/scheduling or file information for specific events. Non-broadcast or VOD services are described here as well.

The Content Element contains the metadata specific to an individual piece of content. This content MAY be an individual show, segments or fragments of a show or aggregates of many shows such as a series.

4.1.3 Structure of Elements

The following six charts depict the structure of the OpenEPG metadata elements.

CHART 1**OpenEPG(82)**

Root element of an Open EPG XML data structure

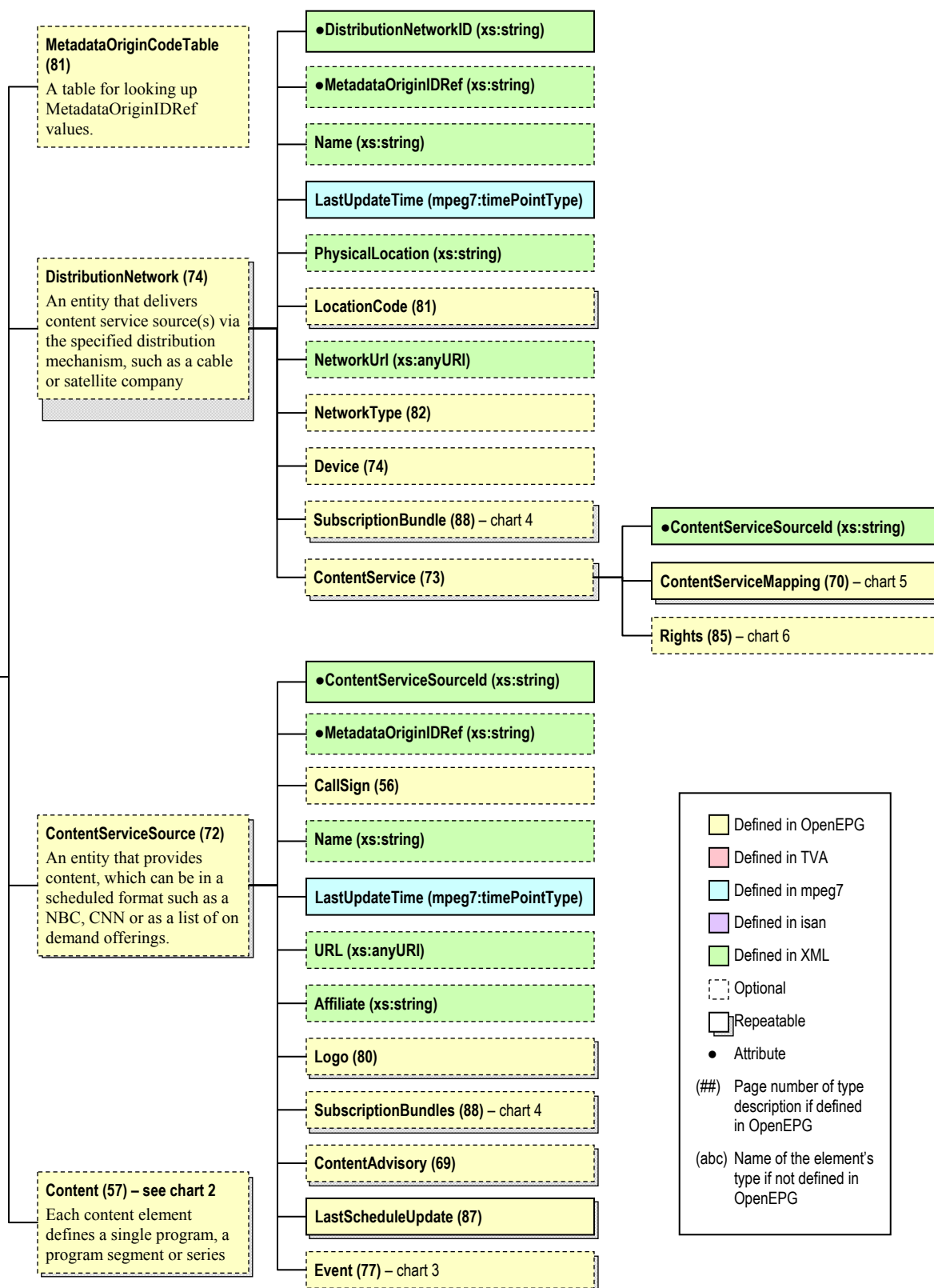


CHART 2

Content (57)

Each content element defines a single program, a program segment or series (from chart 1)

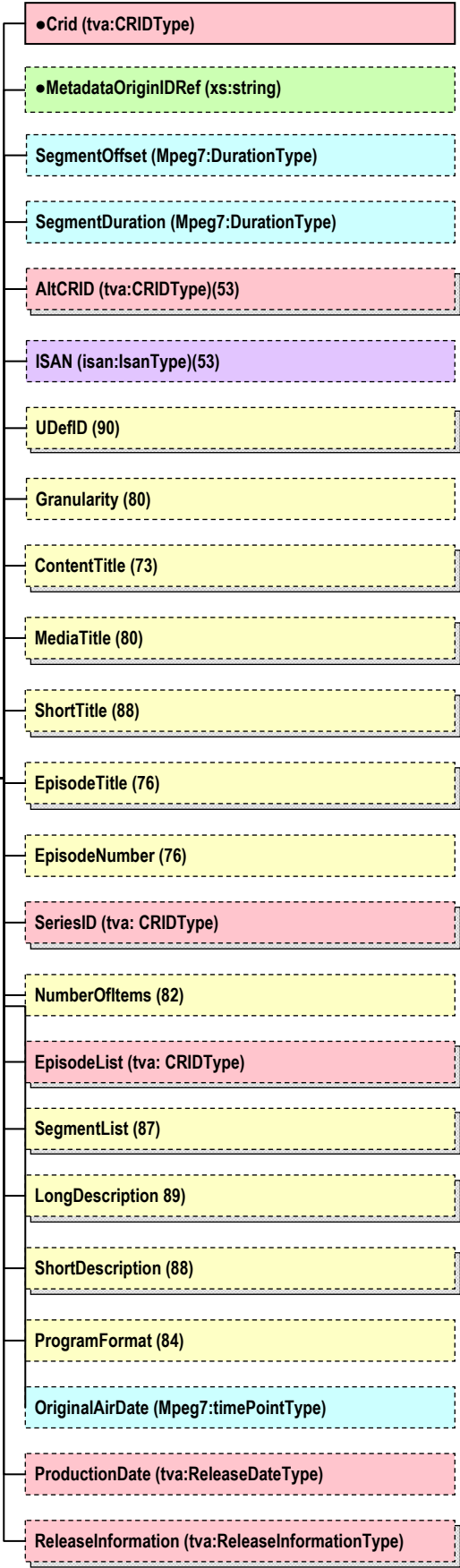


CHART 2 (continued)**Content (57)**

Each content element defines a single program, a program segment or series (from chart 1)

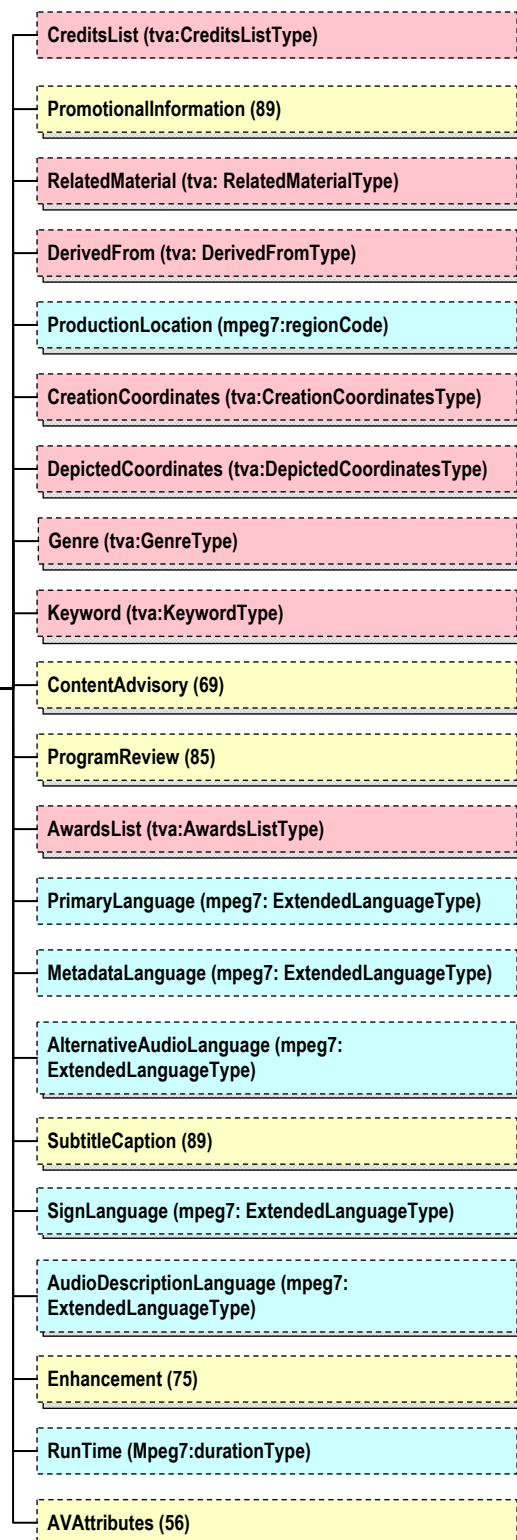


CHART 3

Event (77)
Each Event describes an instance of a piece of content available in a broadcast schedule or as a VOD offering. (from chart 1).

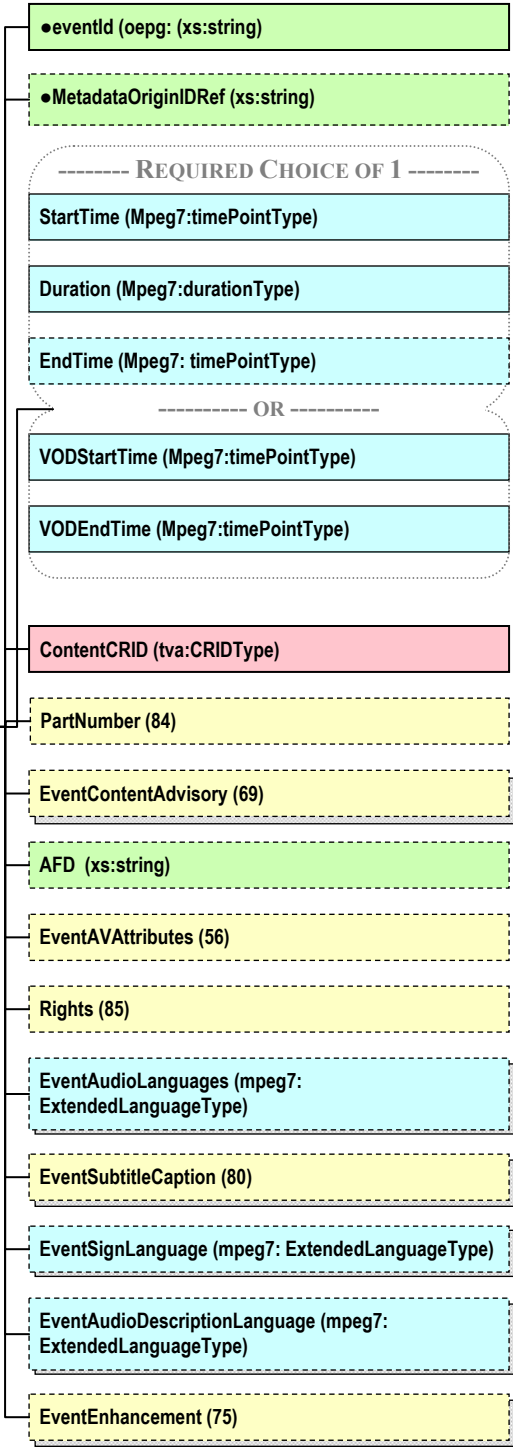


CHART 4**SubscriptionBundle (88)**

Describes bundles available on a distribution network including whether the user is subscribed.

•subscribed (xs:boolean)

•distributionNetworkId (xs:string)

SubscriptionBundleType (xs:string)

CHART 5**ContentServiceMapping (71)**

The location from which content is received, either a broadcast channel designation or internet address, depending on the ContentServiceType

ContentServiceDistributionMethod (69)

----- REQUIRED CHOICE OF 1 -----

Channel (xs:string)

MinorChannel (xs:string)

----- OR -----

URLSource (xs:anyURI)

From (xs:date)

To (xs:date)

ServiceAVAttributes (56)

ServiceAudioLanguages (79)

ServiceSubtitleCaption (89)

ServiceSignLanguages (79)

ServiceAudioDescriptionLanguage (79)

CHART 6**Rights (85)**

User's rights has to view and use content. This information is for user reporting only.

•MetadataOriginIDRef (xs:string)

Blackout (xs:boolean)

PPV(84)

Purchased (xs:boolean)

Price (xs:string)

RecordRights (85)

Play (xs:integer)

RightsText (xs:string)

Copy (xs:integer)

RightsXML (86)

Move (xs:integer)

4.1.4 List of Data Elements

All OpenEPG data elements are defined as XML Types. OpenEPG defines some of these Types while others are taken from the Namespaces that have been defined by other organizations. The full descriptions of each Type and Element are included in Annex 5.1.

4.1.5 Externally Defined Data Types

This standard uses XML Types defined in other standards, including TVA [5], MPEG-7 [18], and ISAN [14]. Users are directed to the original specifications for the precise definitions of these Types.

mpeg7:durationType – Based on the ISO 8601 time formats the format allows for days, hours, minutes, seconds and milliseconds. Its definition is found at ISO/IEC 15938-5: 6.4.4 [19]

mpeg7:ExtendedLanguageType - The Language type with an optional attribute of “original”.

mpeg7:Name - Code string in an mpeg;ControlledTermUseType.

mpeg7:ParentalGuidanceType - The mpeg7 type that defines Parental Ratings.

mpeg7:regionCode – A valid regionCode defined in ISO/IEC 15938-5: 5.6.4 [19]

mpeg7:TextualType – A string with the optional attribute of language field

mpeg7:timePointType – The definition is found at ISO/IEC 15938-5: 6.4.3 [19]

The tns TVAnytime types are defined in ETSI TS 102 822-6-1 V1.2.1 (2005-03).[8]

tns:BinaryPredicateType – Container for Binary Boolean operations

tns:PredicateBagType - Container for Query expressions.

tns:UnaryPredicateType – Container for Unary Boolean expressions.

The tva TVAnytime types are defined in ETSI TS 102 822-3-1 V1.3.1 (2006-01).[7]

tva:AudioAttributesType – A set of audio characteristics for the specified program.

tva:AwardsListType – The list of awards and/or award nominations for the specified program.

tva:BitRateType - The bit rate for a content item. The data value associated with this type defines the average bitrate if the bitrate is variable.

tva:CRIDType - A type to represent a CRID as a URI reference.

tva:ControlledTermType - To make a reference to a Controlled Term. In addition the "Name" and "Definition" of the term can optionally be included. If included and the referenced list of controlled terms are not available, the inline description can be used, otherwise the appropriate controlled term list SHOULD be used to obtain the definitive "Name" and "Definition".

tva:CreationCoordinatesType - A complex type that defines where and when the content was created.

tva:CreditsListType - A list of credits for the specified program.

tva:DepictedCoordinatesType - The location and date depicted within the content.

tva:DerivedFromType - A complex type used to indicate that a program has been derived from a program or a program concept (see ProgramGroupTypeType). It instantiates BaseMemberOfType and is instantiated by DerivedFrom.

tva:GenreType - A data type for specifying a genre for the program.

tva:KeywordType - A data type for specifying a keyword associated to a program.

tva:MediaReviewType - A data type for content reviews

tva:RelatedMaterialType - A complex type that refers to other media assets or segments that are related to the AV content (e.g. program) that is described.

tva:ReleaseDateType - The date when the content was released.

tva:ReleaseInformationType - The Release Information for the content.

tva:VideoAttributesType - A set of elements that describe video characteristics.

4.1.6 Expansion of Element Types

All OpenEPG data elements are defined as XML Types and are designed to be expandable. These Types can be used as base Types within expanded definitions.

4.1.7 OpenEPG Namespace

The OpenEPG Description Scheme (DS) defined in this document has been developed with the OpenEPG namespace defined as:

```
xmlns:oepg="urn:ce:cea-2033:OpenEPG:2006
```

The “oepg” namespace identifier (NID) has been registered with IETF.

The actual schema for OpenEPG is found in file:

OpenEPG-V1.xsd.

Other namespaces referenced by OpenEPG are:

```
xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:mpeg7="urn:mpeg:mpeg7;schema:2001"
```

```
xmlns:tva="urn:tva:metadata:2004"
```

```
xmlns="http://www.isan.org/ISAN"
```

4.1.8 OpenEPG Transport Namespace

The OpenEPG Transport Description Scheme (DS) defined in this document has been developed with the OpenEPG transport namespace defined as:

```
xmlns:oepgtrp="urn:ce:cea-2033:OpenEPGTrp:2006
```

The “oepgtrp” namespace identifier (NID) has been registered with IETF.

The actual schema for OpenEPG Transport is found in file:

OpenEPG-Transport-V1.xsd.

Other namespaces referenced by OpenEPG Transport are:

```
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:mpeg7="urn:mpeg:mpeg7;schema:2001"
xmlns:tva="urn:tva:metadata:2004"
xmlns="http://www.isan.org/ISAN"
```

4.2 Communications

To facilitate the accurate and secure transmission of information from an OpenEPG MSP to an OpenEPG client, transactions SHALL take place on four levels within the communication protocol stack.

1. Discovery of network devices and MSPs.
2. Security, validity and authentication.
3. Request and retrieval of data elements.
4. Notification of the availability of new or updated data elements to clients.

OpenEPG MSPs can be accessed from a WAN (Internet) and within a LAN (Home Network). The requirements and functionality are similar for both environments.

Note: The data-access protocols defined by OpenEPG are designed to be used over bidirectional TCP/IP links using SOAP [21] methods for the low-level wrapper functions. This does not preclude the use of other low-level methods for the actual transfer of data. It is expected that much of the data may be transmitted to home servers by one-way broadcast methods in proprietary formats such as DVB SI or ATSC PSIP. In addition, alternatives exist for two-way communications where data requests are made through a wired (or wireless) IP connection, and the response is sent via an encapsulated broadcast method such as DVB DSMCC or similar ATSC methods. The definition of such alternative transmission methods is beyond the scope of this standard.

4.2.1 Discovery Functions

Discovery consists of WAN Discovery (MSPs and their capabilities) and LAN Discovery (devices and content). Discovery methods vary for WAN- or LAN-based sources.

4.2.1.1 WAN Discovery

Specific methods to access a WAN source are not defined in this standard. Two common methods are noted

4.2.1.2 Referencing a UDDI Directory

A directory service MAY be used to identify various OpenEPG metadata services. The definition of such a directory service is beyond the scope of this standard, but it might be implemented as the UDDI method of the TV-Anytime specification ETSI TS 102-822-6-2 V1.3.1 [9]. If the UDDI method is used, then the Service Description might be included in the UDDI Directory. Whether or not UDDI is supported, an OpenEPG MSP SHALL support the SOAP [21] Method `get_Capabilities` as described in Section 4.2.3.3.1.

4.2.1.3 Referencing a WSDL Implementation

The user/device might specify a URL that points to the MSP's WS-Inspection file, The `WS_Inspection` file points to a Web Services Description (WSDL) document, which contains the available methods (function calls) and their location. Further information can be found in the TV-Anytime specification ETSI TS 102-822-6-2 V1.3.1 [9].

The following text illustrates a WS-Inspection file (WSIL) that references an OpenEPG metadata provider with a `get_Data` port.

Source: <http://example.com/inspection.wsil>

```
<?xml version="1.0" encoding="UTF-8"?>
<inspection xmlns="http://schemas.xmlsoap.org/ws/2001/10/inspection/"
  xmlns:wsilwsdl="http://schemas.xmlsoap.org/ws/2001/10/inspection/wsdl/">
  <service>
    <description referencedNamespace="http://schemas.xmlsoap.org/wsdl/"
      location="http://example.com/TV_week/oepeg_TV_week.wsdl" >
      <wsilwsdl:reference>
        <wsilwsdl:implementedBinding xmlns:ns1="http://example.com/TV_week/oepeg_TV_week.wsdl"
          ">ns1:get_Data_SOAP</wsilwsdl:implementedBinding>
        </wsilwsdl:reference>
      </description>
    </service>
  </inspection>
```

The following is an example of a WSDL referenced in the above WSIL.

Source: http://example.com/TV_week/oepeg_TV_week.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions targetNamespace="urn:openEpg" xmlns:oens="urn:openEpg" xmlns:oepeg="urn:OpenEPGWebServices"
  xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <types>
    <xsd:schema>
      <!-- Import schema for OpenEPG transport types -->
      <xsd:import namespace="urn:OpenEPGWebServices" schemaLocation="OpenEPG_transport.xsd"/>
    </xsd:schema>
  </types>

  <!-- Input/Output Messages -->
```

```

<message name="get_Data_SOAPRequest">
  <part name="get_Data" element="oepg:get_Data"/>
</message>
<message name="get_Data_SOAPResponse">
  <part name="get_Data_Result" element="oepg:get_Data_Result"/>
</message>
<message name="get_Capabilities_SOAPRequest">
  <part name="get_Capabilities" element="oepg:get_Capabilities"/>
</message>
<message name="get_Capabilities_SOAPResponse">
  <part name="get_Capabilities_Result" element="oepg:get_Capabilities_Result"/>
</message>
<message name="soap_Fault_Message">
  <part name="soap_Error" type="xsd:string"/>
</message>
<!-- Define operations on ports -->
<portType name="get_Data_TV_Week">
  <operation name="get_Capabilities">
    <input message="oens:get_Capabilities_SOAPRequest"/>
    <output message="oens:get_Capabilities_SOAPResponse"/>
    <fault name="soap_Fault" message="oens:soap_Fault_Message"/>
  </operation>
  <operation name="get_Data">
    <input message="oens:get_Data_SOAPRequest"/>
    <output message="oens:get_Data_SOAPResponse"/>
    <fault name="soap_Fault" message="oens:soap_Fault_Message"/>
  </operation>
</portType>
<!-- bindings: define how SOAP/HTTP is used -->
<binding name="get_Data_SOAP" type="oens:get_Data_TV_Week">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="get_Capabilities">
    <soap:operation soapAction="get_Capabilities"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="soap_Fault">
      <soap:fault name="soap_Fault"/>
    </fault>
  </operation>
  <operation name="get_Data">
    <soap:operation soapAction="get_Data"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="soap_Fault">
      <soap:fault name="soap_Fault"/>
    </fault>
  </operation>
</binding>
<!-- OpenEPG Web Service -->
<service name="OEPGThisWeek">
  <port name="get_Data_TV_Week_Service" binding="oens:get_Data_SOAP">
    <soap:address location="http://example.com/tv_week"/>
  </port>
</service>

```

```

</service>
</definitions>

```

4.2.1.4 LAN Discovery

LAN Discovery is not addressed in the initial release of this standard.

4.2.2 Security Validity and Authentication Functions

Authentication for OpenEPG MSPs accessed on a WAN is accomplished through the HTTP Digest Access Authentication mechanism defined in IETF RFC 2617 [11]

A username and password are issued in response to the *Authenticate* challenge from the OpenEPG MSP. Table 1 shows the properties that are defined for the authentication header:

Table 1: HTTP Digest Access Authentication mechanism parameters

Attribute	Value
algorithm	MD5
realm	OpenEPGServiceRealm
qop	Auth

All further communication between the client and the MSP SHALL use SOAP over HTTP protocols. Transport Layer Security (TLS) MAY be used if secure communications are required.

Once authentication has been established between the MSP and the client, the MSP SHOULD determine the user's rights to any information or content. This standard does not specify the methods for the establishment or management of these rights.

4.2.3 MetaData Access Functions

4.2.3.1 Overview

The OpenEPG data structure and access methods have been designed to make the development of client and server applications straightforward. The XML document structure eliminates redundancies and makes data transfer as efficient as possible.

The OpenEPG standard makes use of the TV-Anytime query syntax and the SOAP `get_Data` method to request OpenEPG metadata from an MSP. TVA uses names from a table of aliases in TVA queries. OpenEPG does not have a table of aliases and requires that the entire path of a data element be specified in queries. (See examples below in Sections 4.2.3.3.2 and 4.2.4)

A set of predefined Query Templates (Request and Response) have been developed, which are designed to simplify the exchange of data between clients and MSPs. These templates set bounds on the types and complexity of queries that SHALL be supported by OpenEPG servers,

by specifying typical subsets of data to be returned. MSPs MAY vary in their support of various services and templates.

4.2.3.2 Operation of a Data Request

Once the URL of an OpenEPG service has been identified, the procedure for getting data is generally the same in a LAN environment and a WAN environment. The client MAY get the capabilities of the service with the `get_Capabilities` request. This tells the client what kinds of queries are supported. Some services MAY only support a limited number of Request or Response Templates (formats). This may be especially true of services that provide only a very small subset of data such as content reviews. Other services may have a full general purpose database engine that can support the full universe of queries that are possible on the OpenEPG data set. Services that support full Boolean queries SHALL indicate that by setting the “**supportAdvancedQueries**” attribute in the `get_Capabilities` response. Requests to services that do not support advanced Queries SHALL be made according to the templates listed in the “**AvailableRequestTemplates**” of the `get_Capabilities` response.

In addition to building the request an application SHALL also specify the data elements that it wants returned. This MAY be done in one of two ways or a combination of them. For many typical queries the data required MAY be specified in one of the Response Templates. These list the data elements that MAY be returned if the service has that particular data element. The application MAY also specify the data elements individually that it wishes returned. For some applications there MAY be a template that specifies most but not all of the elements required. In this case the template MAY be used along with addition fields specified individually. In all cases the order of the data elements SHALL follow the element order in the OpenEPG data schema.

4.2.3.3 Queries

OpenEPG queries can be grouped into 2 categories:

- **get_Capabilities** (described in Section 4.2.3.3.1) – functions that request the capabilities of the MSP.
- **get_Data** (described in Section 4.2.3.3.2) – functions that request the actual metadata

The `get_Data` function is further divided into two sub categories.

- **General Queries** – can ask any question that the syntax will allow
- **Templates Queries** – are limited by the Query Template to a specific set of questions that can be asked

Both General Queries and Template Queries MAY use a Response Template to indicate the list of data elements to return.

An MSP MAY limit the types of queries that it supports, the client can first determine the response capabilities of the MSP. The client can then issue a query for the desired metadata in the appropriate manner.

These queries are described in the following subsections.

4.2.3.3.1 **get_Capabilities: Establishing the queries that an MSP supports**

The get_Capabilities command is used to determine which Request or Response templates are supported by an MSP, and whether it additionally supports General queries. (See Section 4.2.3.3.2 for a description of Template and General Queries).

A get_Capabilities query follows this structure:

```
POST /mpeg/metadata-service HTTP/1.0
Host: www.openepg.com
Content-Length: xxxx
Content-type: text/xml; charset="utf-8"
SOAPAction: "get_Capabilities"

<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns=http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <get_Capabilities xmlns="http://www.openepg.com/...">
  </Body>
</Envelope>
```

The response to the get_Capabilities query is an XML document describing the query capabilities of the server (General Queries or a list of specific Template Queries).

The document has the following structure:

```
<complexType name="get_Capabilities_ResultType">
  <sequence>
    <element name="Name" type="String" minOccurs="0"/>
    <element name="Description" type="mpeg7:TextualType" minOccurs="0" maxOccurs="unbounded"/>
    <element name="AvailableRequestTemplates" type="mpeg:AvailableRequestTemplatesListType"
minOccurs="1" maxOccurs="unbounded"/>
    <element name="AvailableResponseTemplates" type="mpeg:AvailableResponseTemplatesListType"
minOccurs="1" maxOccurs="unbounded" />
  </sequence>
  <attribute name="supportAdvancedQueries" type="boolean" use="required"/>
  <attribute name="serviceVersion" type="unsignedInt" use="required"/>
</complexType>
```

Each MSP MAY limit the types and sophistication of the queries it supports, but an MSP SHALL support one or more Query Templates. (See Section 4.2.4). Support of these templates SHALL be indicated in the get_Capabilities response.

An MSP SHALL indicate in the get_Capabilities response whether or not it also supports General Queries by setting the "supportAdvancedQueries" flag.